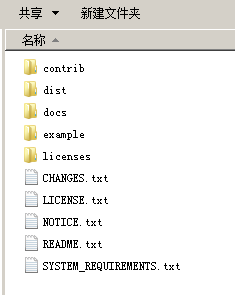
1、下载solr应用

下载地址：<http://lucene.apache.org/solr/downloads.html>

建议下载solr-4.9.1，解压后



1. 将解压后的solr-4.9.1文件中的example/webapps 下的solr.war拷贝到tomcat下的webapps下解压缩成solr
2. 将解压后的solr-4.9.1文件中的example文件夹下的solr拷贝你电脑的路径中，不要有中文，然后将名字修改为solr\_home
3. 修改tomcat中webapp下solr文件中WEB-INF目录下的web.xml文件，在开头添加以下一段内容

<env-entry>

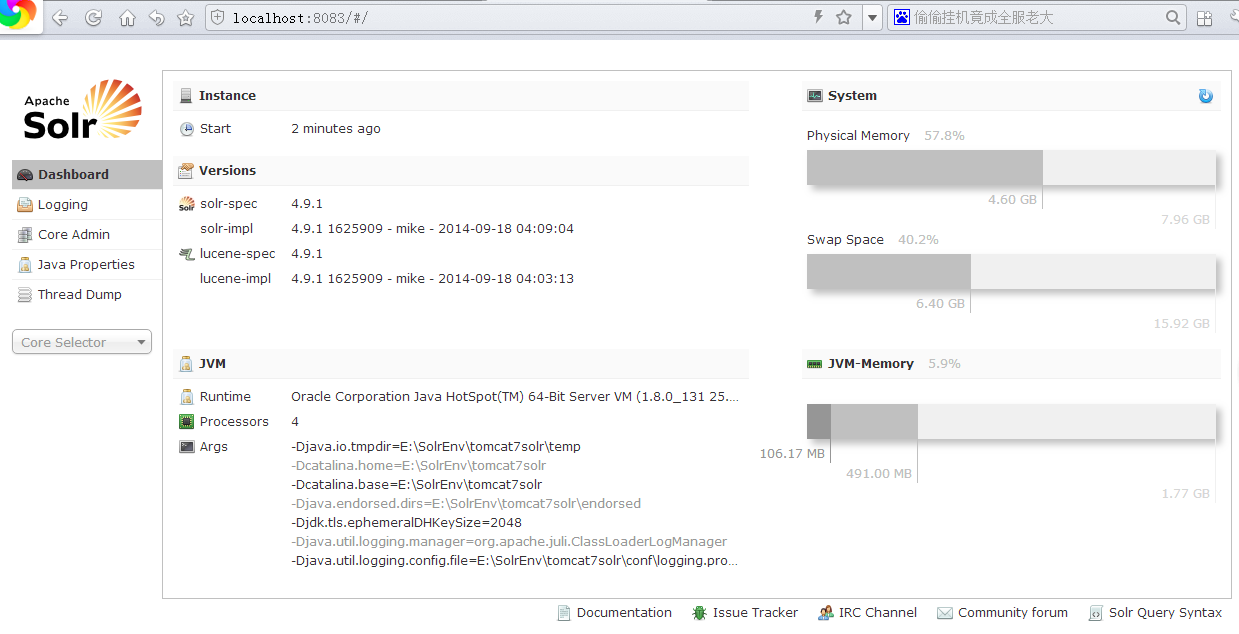
<env-entry-name>solr/home</env-entry-name>

<env-entry-value>你自己配置的solr\_home的位置</env-entry-value>

<env-entry-type>java.lang.String</env-entry-type>

</env-entry>

1. 将solr-4.9.1文件中example/lib/ext中的jar包拷贝到Tomcat下面的lib目录中
2. 启动solr服务器（也就是启动tomcat服务器），本地地址栏访问localhost:端口/应用名 看到如下结果证明启动成功



1. 在solr\_home目录下的新建core，只需要复制案例中的collection1目录，重命名为hotel，并修改配置文件core.properties中的值为对应的目录名即可（即name=hotel）。
2. 重启tomcat访问查看core selector中刚才配置的core，至此环境搭建成功
3. 实现数据的配置
4. 拷贝数据库驱动包（mysql-connector-java-5.1.18.jar）放到tomcat的lib目录下
5. 新建core（8中已经实现，这里就可以省略）
6. 打开hotel中conf文件中的solrconfig.xml，在里面添加一个插件指令实现数据导入

<requestHandler name="/dataimport" class="org.apache.solr.handler.dataimport.DataImportHandler">

<lst name="defaults">

<str name="config">data-config.xml</str>

</lst>

</requestHandler>

1. 在hotel的conf目录下创建data-config.xml文件

<?xml version="1.0" encoding="UTF-8" ?>

<dataConfig>

<dataSource tpye="JdbcDataSource" driver="com.mysql.jdbc.Driver" url="jdbc:mysql://127.0.0.1:3306/itripdb" user="root" password="root"/>

<document name="hotel\_doc">

<entity name="hotel" pk="id" query="select id,hotelName,address,details from itrip\_hotel">

<field column="id" name="id"/>

<field column="hotelName" name="hotelName"/>

<field column="address" name="address"/>

<field column="details" name="details"/>

</entity>

</document>

</dataConfig>

dataSource:数据库数据源

entity：一张表对应的实体，pk就是主键，query就是查询语句

filed：对应一个字段，column是数据库里的column名，后面的name属性对应着solr的filed的名字

1. 找到hotel的conf目录下的schema.xml文件
2. 删除所有的<field>、< dynamicField>和< copyField>
3. 添加以下俩个节点

<field name="\_version\_" type="long" indexed="true" stored="true"/>

<field name="text" type="text\_general" indexed="true" stored="false" multiValued="true"/>

name:属性的名称

type:属性的类型

indexed：是否创建索引（true:是、false:否，如果不需要存储相应字段值，尽量设为false）

multiValued：是否有多个值，比如说一个用户的所有好友id。（对可能存在多值的字段尽量设置为true，避免建索引时抛出错误）

stored：是否存储原始数据（true:存储 false:不存储）

1. 添加索引字段，这里的每个field的name要和data-config.xml里的entity的filed的name一样，一一对应。
2. 如果将字段类型改成long的时候需要修改一下，将solr\_home下面hotel/conf/solrconfig.xml文件中的以下内容注释掉，否则会报错。

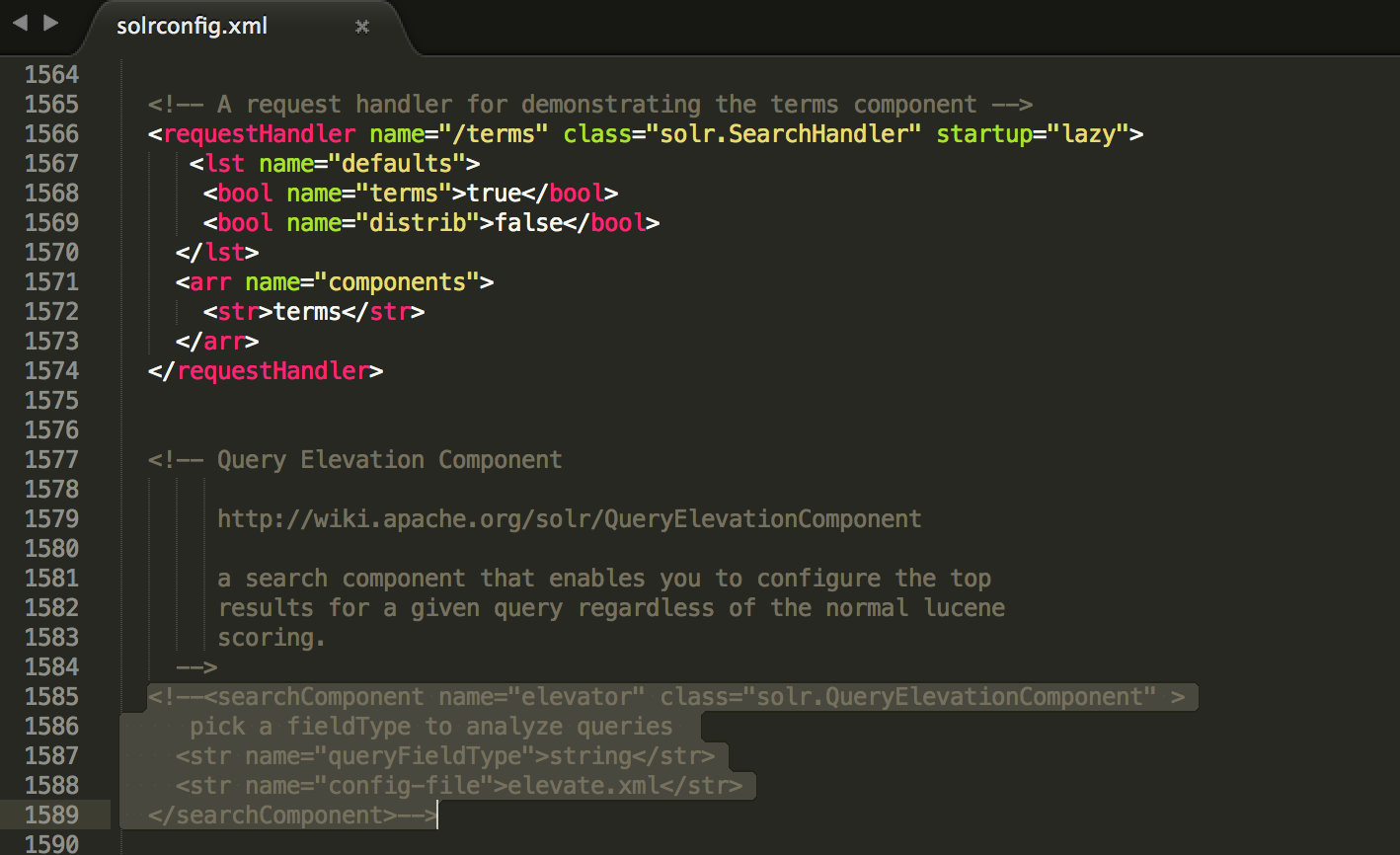
<searchComponent name="elevator" class="solr.QueryElevationComponent" >

pick a fieldType to analyze queries

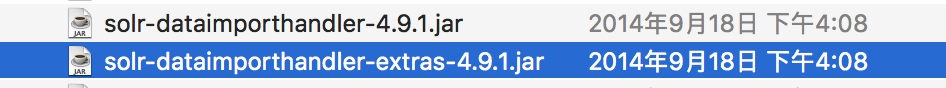
<str name="queryFieldType">string</str>

<str name="config-file">elevate.xml</str>

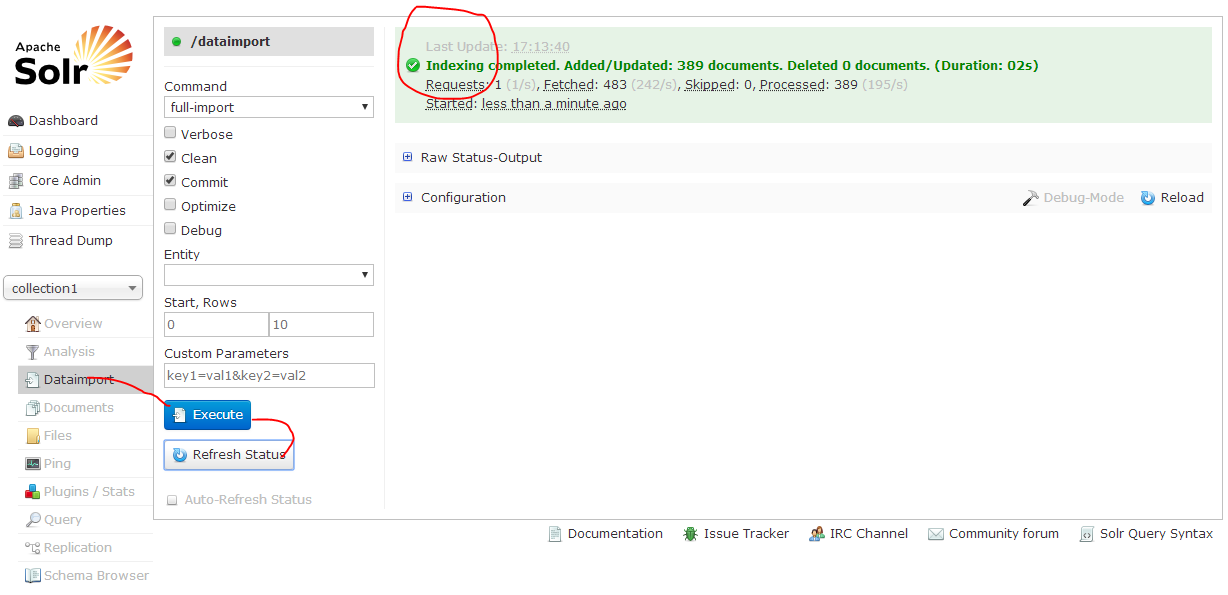
</searchComponent>



1. 将solr-4.9.1文件中dist目录中的俩个与文件导入相关的jar包拷贝到Tomcat下面webapps下面中的solr下的WEB-INF/lib目录中



1. 启动solr服务器，出现以下情况就表示配置成



1. 实现数据增量更新
2. 将资料中提供的apache-solr-dataimportscheduler.jar拷贝到tomcat中的webapps/solr/WEB-INF/lib下
3. 在solr\_home文件夹下新建conf文件，并新建名为dataimport.properties配置文件，添加以下内容

# to sync or not to sync

# 1 - active; anything else - inactive

syncEnabled=1

# which cores to schedule

# in a multi-core environment you can decide which cores you want syncronized

# leave empty or comment it out if using single-core deployment

syncCores=hotel

# solr server name or IP address

# [defaults to localhost if empty]

server=localhost

# solr server port [defaults to 80 if empty]

port=8080

# application name/context

# [defaults to current ServletContextListener's context (app) name]

webapp=solr

# 增量索引的参数

# URL params [mandatory]

# remainder of URL

params=/dataimport?command=delta-import&clean=false&commit=true

# 重做增量索引的时间间隔

# schedule interval

# number of minutes between two runs

# [defaults to 30 if empty]

interval=1

# 重做全量索引的时间间隔，单位分钟，默认7200，即5天;

# 为空,为0,或者注释掉:表示永不重做索引

#reBuildIndexInterval=7200

# 重做索引的参数

reBuildIndexParams=/dataimport?command=full-import&clean=true&commit=true

# 重做索引时间间隔的计时开始时间，第一次真正执行的时间=reBuildIndexBeginTime+reBuildIndexInterval\*60\*1000；

# 两种格式：2012-04-11 03:10:00 或者 03:10:00，后一种会自动补全日期部分为服务启动时的日期

reBuildIndexBeginTime=03:10:00

1. 去tomcat中的webapps/solr/WEB-INF/web.xml添加以下内容

<listener>

<listener-class>

org.apache.solr.handler.dataimport.scheduler.ApplicationListener

</listener-class>

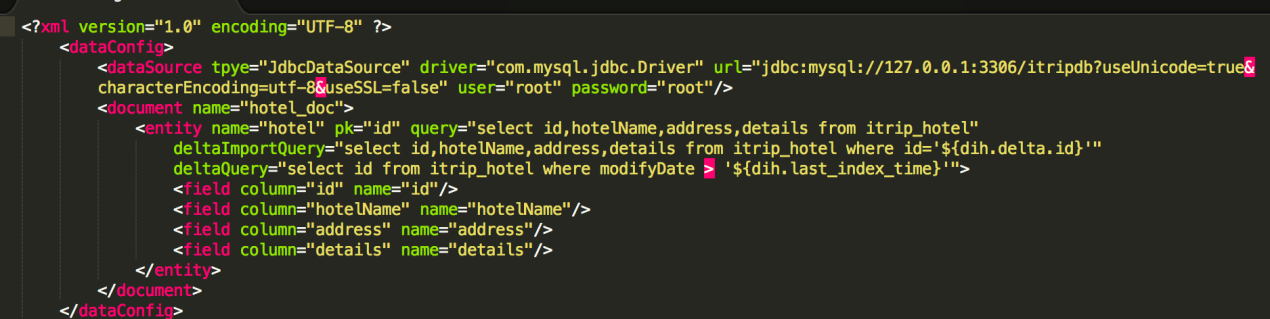
</listener>

1. 修改solr\_home/hotel/conf/data-config.xml配置文件

编写增量更新语句

deltaImportQuery="select id,hotelName,address,details from itrip\_hotel where id='${dih.delta.id}'"

deltaQuery="select id from itrip\_hotel where modifyDate > '${dih.last\_index\_time}'"



e)启动测试

1. 全文检索之IK分词器操作步骤

a)下载IK分词器

拷贝jar（IKAnalyzer2012FF\_u1.jar）到tomcat\webapps\solr\WEB-INF\lib中

b) tomcat\webapps\solr\WEB-INF\下新建classes文件夹，拷贝IKAnalyzer.cfg.mxl和stopword.dic到classes文件夹下

c)在solr\_home/hotel/conf/schema.xml文件中添加iK分词器字段类型

<fieldType name="text\_ik" class="solr.TextField">

<analyzer type="index" isMaxWordLength="false" class="org.wltea.analyzer.lucene.IKAnalyzer"/>

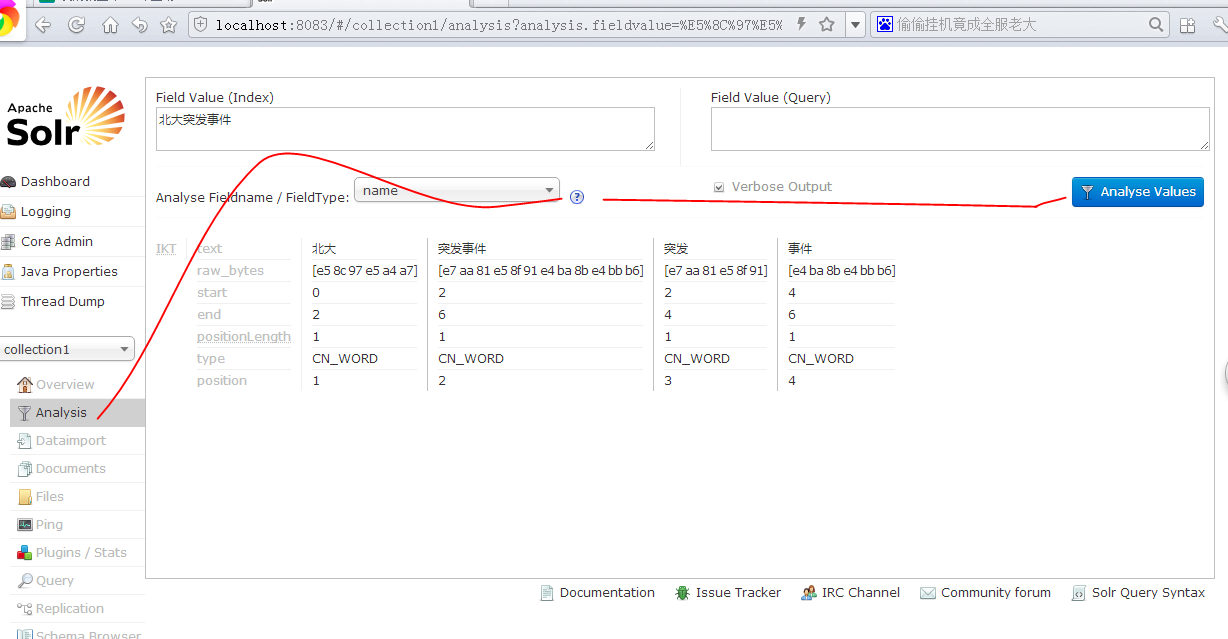
<analyzer type="query" isMaxWordLength="true" class="org.wltea.analyzer.lucene.IKAnalyzer"/>

</fieldType>

1. 配置分词器

在solr\_home/hotel/conf/schema.xml文件中修改需要实现分词器功能的字段类型设置成text\_ik类型

1. 启动测试



1. 多字段查询

在在solr\_home/hotel/conf/schema.xml文件中添加以下内容

<field name="keyword" type="text\_ik" indexed="true" stored="true" multiValued="true"/>

<copyField source="hotelName" dest="keyword" />

<copyField source="address" dest="keyword" />